RESEARCH ARTICLE

Image steganography based on DNA sequence translation properties

Abdullah Ahmed Abdullah[1], Sardar Hasen Ali[1], Ramadhan J. Mstafa[1], Vaman Mohammed Haji[1]

[1]*Department of Computer Science, Faculty of Science, University of Zakho, Duhok, Kurdistan Region, Iraq*

Corresponding author's email: *abdullah.abdullah@uoz.edu.krd*

Received: 24/11/2019 **Accepted:** 14/03/2020 **Available online:** 30/06/2020

ABSTRACT

Digital communication has become a vital part of daily life nowadays. Many applications are using internet-based communication, and here, the importance of security rose to have secure communication between two parties to prevent unauthorized access to sensitive data. These requirements led to several studies in information security that have been done in the past two decades. Cryptography and steganography are the two main methods that are being used for information security. Cryptography refers to techniques that encrypt a message to be sent to a destination using different methods. In contrast, steganography is the science of hiding information from others using another cover message or media such as image, audio, video, and deoxyribonucleic acid (DNA) sequence. This paper proposed a new method to hide information in an image using the least significant bit (LSB) based on DNA sequence. To accomplish this, the proposed scheme used properties of the DNA sequence when codons that consist of three nucleotides are translated to proteins. The LSBs of two pixels from the image are taken to represent a codon and then translate them to protein. The secret message bits are injected into codons before the translation process, which slightly distorts the image and makes the image less suspicious and the hidden message hard to detect. The experimental results indicate the effeteness of the proposed method with a peak signal-to-noise ratio of 57.33 at 0.7 hiding capacity.

Keywords: Image steganography, DNA sequence, Data hiding, Security, LSB.

1. INTRODUCTION

Ever since humans started to communicate with each other remotely, the need for secure communication existed, and it rapidly increases by time. The exchange of information in the internet era has led to a fast increase in the information exchange between the

easier to send information via the internet, it suffers from a major challenge, which is secure data transfer (Comer, 2019). This is all due to the increasing number of hacking and intrusion incidents every year (Amoroso, 1999). To overcome secure communication issues, two main different methods have come to exist and have been used extensively all over the world, which include cryptography and steganography (Wayner, 2009). Cryptography and

tely (Abdo, Sabry, and Ali, 2018; Hussain, Wahab, Idris, Ho, and Jung, 2018). Nonetheless, they can be combined to create more robust techniques, and this makes it harder to break the security (Gupta, Ankur Goyal and Bhushan, 2012). Whereas cryptography includes techniques to encrypt a message to make it noisy data, steganography is techniques that hide a message inside a cover medium without distorting too much information on the used medium to make it less suspicious

Access this article online

DOI: 10.25079/ukhjse.v4n1y2020.pp15-26 **E-ISSN:** 2520-7792

(Mstafa and Elleithy, 2017). Different types of mediums can be used, such as image, audio, video, and DNA sequence (Cheddad, Condell, Curran, and Mc Kevitt, 2010; Djebbar, Ayad, Meraim, and Hamam, 2012; Mstafa, Elleithy, and Abdelfattah, 2017; Shiu, Ng, Fang, Lee, and Huang, 2010). Distorting medium to an extended level may give a hint to steganalysis that such medium contains a hidden message and thus may make it easier for attackers to detect the hidden message (Denemark, Boroumand, and Fridrich, 2016). Different mediums can be used alongside each other to make a more robust and less suspicious medium when hiding a message. A large number of methods have used an image for hiding information, and some researchers used DNA sequence properties to hide data inside the sequence (Shiu et al., 2010).

This paper proposed a robust information hiding method. The method uses least significant bit (LSB) of the cover image to hide data inside an image on the basis of the biological properties of the deoxyribonucleic acid (DNA) sequence with a low modification rate, which makes the cover image less suspicious to attackers. The cracking probability for the proposed method is high, which makes it hard to detect the original message.

The rest of this paper is organized as follows: Section 2 gives the reader an overview of the DNA sequence. Section 3 briefly outlines related works. Section 4 is the presentation of the proposed scheme for hiding a message. Section 5 discusses the analysis of the performance of the presented scheme. The last section has the conclusion of this paper.

2. INTRODUCTION TO DNA SEQUENCE

Understanding how the DNA works and what properties it has is essential to use it as a cover or to use its properties for data hiding. In general, DNA sequences for all living things are made up of four chemical structures, which are referred to as nucleotides. The four nucleotides are adenine (A), thymine (T), cytosine (C), and guanine (G) (Jorde, Carey, and Bamshad, 2016). In some instances, nucleotides may not be of any of these four types, which is then denoted by N as non-labeled nucleotides, and they have no role in protein synthesis (Huang, Chang, and Wu, 2014).

The aforementioned bases of DNA are essential in all biological organisms. Alignment and arrangement of the four types of nucleotides in the DNA sequence are responsible for making different types of proteins that are responsible for activities in all living organs. To make protein, DNA sequence is copied to ribonucleic acid (RNA) in a process called transcription, which is an interim process to make protein (Nussbaum, McInnes, and Willard, 2016). The process is not random but rather depends on some rules called complementary rule of DNA/RNA, and these rules depend on chemical shape and bonds of nucleotides, which consist of A-T, T-A, C-G, G-C (Yurke, Turberfield, Mills, Simmel, and Neumann, 2000). The next process that occurs to make protein is called translation, and during this process, the RNA sequence translates to an amino acid sequence, which is the base of protein synthesis (Nussbaum et al., 2016). The translation process works as follows: every three nucleotides are taken together in the RNA to make one codon, and then different codons code for different types of amino acids. Amino acids undergo multiple processes to make proteins (Jorde et al., 2016). The total number of amino acids is 20, which comes from 61 combinations of codons and three codons that do not code for amino acids and are referred to as STOP codons, which are responsible for ending sequences during the transcription process (De Silva and Ganegoda, 2016). Codons and their corresponding amino acids are shown in Table 1.

Table 1: Codons list with their represented Amino acid

Parameters			Quantity
AAA	Lysine	CAA	Glutamine

AAC	Asparagine	CAC	Histidine
AAG	Lysine	CAG	Glutamine
AAT	Asparagine	CAT	Histidine
ACA	Threonine	CCA	Proline
ACC	Threonine	CCC	Proline
ACG	Threonine	CCG	Proline
ACT	Threonine	CCT	Proline
AGA	Arginine	CGA	Arginine
AGC	Serine	CGC	Arginine
AGG	Arginine	CGG	Arginine
AGT	Serine	CGT	Arginine
ATA	Isoleucine	CTA	Leucine
ATC	Isoleucine	CTC	Leucine
ATG	Methionine	CTG	Leucine
ATT	Isoleucine	CTT	Leucine
GAA	Glutamate	TAA	STOP
GAC	Aspartate	TAC	Tyrosine
GAG	Glutamate	TAG	STOP
GAT	Aspartate	TAT	Tyrosine
GCA	Alanine	TCA	Serine
GCC	Alanine	TCC	Serine
GCG	Alanine	TCG	Serine
GCT	Alanine	TCT	Serine
GGA	Glycine	TGA	STOP
GGC	Glycine	TGC	Cysteine
GGG	Glycine	TGG	Tryptophan
GGT	Glycine	TGT	Cysteine
GTA	Valine	TTA	Leucine
GTC	Valine	TTC	Phenylalanine
GTG	Valine	TTG	Leucine
GTT	Valine	TTT	Phenylalanine

3. RELATED WORK

Over years, many techniques have been developed for hiding information in an image as a cover medium, in which LSB-based hiding is one of the most used techniques (Nag, Choudhary, Basu, and Dawn, 2016; ur Rehman, Liao, Kulsoom, and Abbas, 2015). Even though at first the stego-message bits were directly injected to LSB of image pixels on the basis of the 24 color bits of red, green, and blue (RGB), over time, many researchers have been modifying LSB techniques especially in special domain (Hussain et al.,

2018). Hologram of LSB has been proposed by [LSB+] firstly, and then different authors have worked on the same concept later on (Wu, Dugelay, and Cheung, 2008). The hologram was to add and change some bit to the cover image, where those bits are not from the secret message, and thus, the cover image will have modified bits that either belong to secret message or are some arbitrary bits that will make it hard for attackers and steganalysis to detect the secret message bits. Furthermore, some author has worked on only one or two colors in the spatial domain when specified conditions are met (Cheddad et al., 2010).

DNA has also been used by many researchers for data security (Leier, Richter, Banzhaf, and Rauhe, 2000; Shiu et al., 2010). DNA-based steganography is attracting researchers because of these three properties: 1- Invisibility, hiding information from steganalysis, 2- Hiding capacity, it has a reasonable size to hide message, 3- Consistency, the medium is not altered too much when hiding a message (Hafeez, Khan, and Qadir, 2014). For hiding a message in the DNA sequence, the arrangement of nucleotides is either manipulated or altered. For the ease of the process, the nucleotides are converted from English letters to binary representation, and Leier was the first one who worked on the binary representation of DNA's nucleotides (Leier et al., 2000). Some authors (Abdo et al., 2018; Shiu et al., 2010) have worked on random nucleotides based on some mathematical models; however, altering DNA sequence to an extended level would make DNA sequence suspicious and may give some hints to attackers. Furthermore, the sequence may lose its biological functionality when converted to RNA to make proteins in the future. Hafeez (Hafeez et al., 2014) proposed a technique to hide information inside DNA sequence while preserving the sequence's biological functionality.

4. PROPOSED SCHEME

The proposed method is based on the LSB technique to hide a secret message in a cover image medium. However, unlike the traditional LSB method in which bits of the stego message are injected directly into RGB, the proposed method injects bits of stego message to some selected colors of some pixels, even though the hiding capacity decreases; nonetheless, it makes it harder to detect the message in the cover image. To achieve this goal, the proposed method depends on the DNA sequence properties of the translation and the transcription process, which are the early stages of synthesizing proteins. Because the cover image LSBs are represented by binary for RGB and DNA's nucleotides are represented by four English letters, a mechanism should be established to map between the two representations. Thus, the proposed method will use the binary representation of nucleotide letters according to Table 2.

Table 2: The binary representation of nucleotides

Nucleotide	Binary representation
A	00
G	01
C	10
T	11

In addition to LSB for data hiding, the mapping (translation process on DNA sequence) table for amino acids is needed for the proposed method. Table 3 shows the mapping used for hiding in the proposed method. Total combinations of codons are 64 codons, which came from three letters

combination of the four nucleotides (four to the third power). The 64 codons are linked to 20 amino acids and the stop codons.

However, the mapping between codons and amino acids is not uniform, which means that not all amino acids have the same number of codons, and thus, some amino acids are represented by as few as one codon, whereas some are represented by as many as six codons. Nonetheless, some patterns can be found from amino acids that are making the same amino acid. Obvious patterns are for the case of two codons when representing an amino acid; they start with the same letters except for that last one, which ends with either (A and G) or (C and T). This means that whenever the A is altered to G or vice versa, the resulting amino acid will not change, the same role applies to C and T as colored blue in Table 3. For the case of four codons that represents the same amino acid, the first two letters are the same, and the third letter is the only one that changes. Irrespective of the value of the third letter, the result is always the same amino acid. The case of six codons per amino acid is nothing more than a combination of two and four cases. For the case in which only one codon represents an amino acid, nothing can be altered because the change will result in a different amino acid. The case of three codons also has two codons that can exchange letters between (A and G) or (C and T), whereas the third codon cannot be changed because it will result in a different amino acid. Codons in Table 3 are colored using three colors, where blue means only (A and G) or (C and T) can be used interchangeably to produce same amino acid, green is used for four codon that produce same amino acid, for which the last letter does not affect the result of the resulted amino acid, and finally, red is used for codons where any change in the codon will result in a different amino acid. Two properties, which are the variance on the number of codons and the flexibility of nucleotide that can make the same amino acid, are used in this work to hide a secure message inside a cover image.

Table 3: Codons to Amino acids exchange dictionary

Amino acid	Codons
Lysine	AAA, AAG
Asparagine	AAC, AAT
Glutamine	CAA, CAG
Histidine	CAC, CAT
Glutamate	GAA, GAG
Aspartate	GAC, GAT
Tyrosine	TAC, TAT
Cysteine	TGC, TGT
Phenylalanine	TTC, TTT
Threonine	ACA, ACG, ACC, ACT
Proline	CCA, CCG, CCC, CCT
Alanine	GCA, GCG, GCC, GCT
Glycine	GGA, GGG, GGC, GGT
Valine	GTA, GTG, GTC, GTT
Arginine	AGA, AGG, CGA, CGG, CGC, CGT
Serine	AGC, AGT, TCA, TCG, TCC, TCT
Leucine	TTA, TTG, CTA, CTG, CTC, CTT
Isoleucine	ATC, ATT, ATA
STOP Codons	TAA, TAG, TGA

4.1 Data hiding method

In the data hiding process, the LSBs of the cover image are taken to hide a message inside it. The hiding process does not directly inject the value to the LSBs but rather depends on the value of LSBs that are next to each other in image pixels. For the embedding process, each time, two pixels are taken from an image in sequential order and their LSB for RGB values are taken. This process will result in the creation of a string of six bits (R1

G1 B1 R2 G2 B2), where the R1G1B1 belongs to the first pixel and the R2G2B2 belongs to the second pixel. These six bits are converted to a three-letter codon on the basis of the binary representation from Table 2, and then, the codon value is checked against the Table 3 codons. When comparing values with codons in Table 3, three different results can be found based on the color used in the table as follow: first, when the codon value matches a red color on the table, then the value remains unchanged, and the two pixels will not be used for hiding. Second, if the value matched one of the blue colored codons, then the last bit can be used for the hiding process because according to Table 2, when the last bit is changed, the codon last letter alters from A to G and vice versa when last letter equal to A or G; otherwise, it alters from C to T and vice versa for C and T. According to Table 3, these changes will preserve the same amino acid as result. Thus, the last bit which mapped to the blue color in the second pixel is used for hiding purposes. Third, if the value of the codon matches with a green value in the Table, then the last two bits can be used for hiding since according to Table 3 the last letter green color changed to whatever possible nucleotides the resulted amino acid remains unchanged. Thus, the last two bits which are mapped to green and blue values of the second pixel are used for hiding purpose. In short, red color codons in Table 3 cannot be used for hiding, blue color codons can hide one bit and green colored codons can hide two bits. Furthermore, bits from the message that is to be hidden in the cover image is XORed with the seven most significant bits instead of directly injecting the bit into the image. The data hiding algorithm works as follow:

Data Hiding Algorithm:

Input:

M: stego message

IMG: cover image

Output:

SIMG: image with stego message

Method:

1. Initialize the amino acid dictionary based on Table 3.
2. Convert the message M to binary message BM.
3. Take two pixels from IMG in sequential order.
4. Get values of LSBs for RGBs from the two pixels of step 3 in sequential order R1G1B1R2G2B2.
5. Map the six bits value of step 4 to three-letter combinations of nucleotides (codons) based on Table 2 which make one codon.
6. Check a match for the codon from step 5 in Table 3 and compare the value with the matched color group:
    - If the color is red, go to step 7 and continue.
    - If the color is blue, hide one bit of BM by using the XOR gate with seven most significant bits of the second pixel's blue color.
    - If the color is green, hide two bits of BM using the XOR gate with seven most significant bits of second pixel's green and blue colors, respectively.
7. Save the result in LSB of the corresponding Byte.
8. If BM is not empty, go to step 3. Otherwise, go to the next step.
9. Add eight bits with value zero as a stop condition and hide them as normal stego message.
10. The process is completed and the result is stego image SIMG.

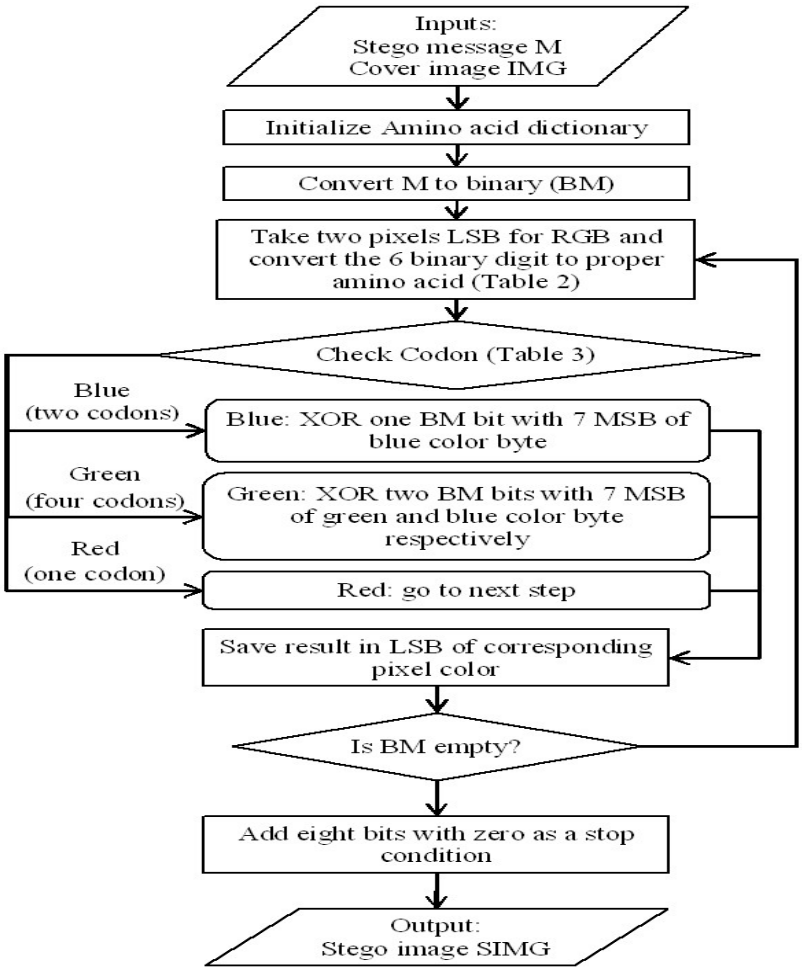

Figure 1: Data hiding phase flow chart

The example of the data hiding scheme works as follows: suppose data type that is to be sent is text, and suppose the message is "hello." The first thing is to convert the message to binary form using ASCII code, and thus, the binary message is equal to "01101000 01100101 01101100 01101100 01101111." The next step is choosing an image as cover, and then the algorithm takes LSBs of RGB in sequence for the first two pixels in the cover image, suppose the value for LSBs are 011011. The next step is to convert this binary sequence to nucleotides; according to Table 2, the generated value will equal "GCT." Then compare this sequence on Table 3 and find codons that are grouped with this sequence, if any. In the given example value, it happens to have four codons in the same group (GCA, GCG, GCC, GCT). Based on the algorithm, if the last letter has been exchanged with any other letters, then the produced codon will be in the same group. For that one letter which represents two bits can be altered using XOR logic gate. Thus, the first two bits in the binary message are taken and the last two bits in the LSB sequence are replaced with the two bits of binary message, and thus, the second pixel of cover image will be altered. The same process will continue until the whole binary message is embedded in the cover image.

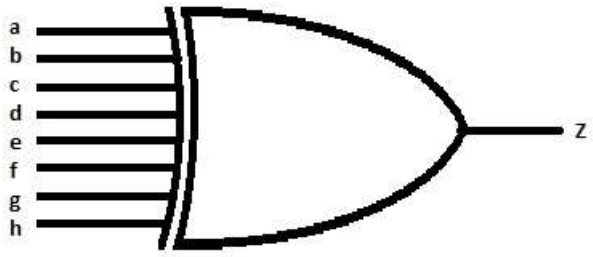

Figure 2: the XOR gate with eight inputs

4.2. Recovery Method

In this phase, the receiver gets the stego image that contains the hidden message that can be recovered in the reverse of the data hiding algorithm. No key needed to recover the stego message from the image. However, all procedures that have been used in the hiding algorithm should be followed, which includes binary representation of nucleotides from Table 2, mapping nucleotides from Table 3, the use of colors, number of bits for hiding process, and number of pixels that are taken each time. The algorithm for the recovery process works as follow:

Data Recovery Algorithm:

Input:

SIMG: image with stego message

Output:

M: stego message

Method:

1. Initialize the amino acid dictionary based on Table 3.
2. Take two pixels from stego image SIMG in sequence order.
3. Get values of LSBs for RGBs from the two pixels of step 3 in sequential order $R_1G_1B_1R_2G_2B_2$.
4. Map the six bits value of step 3 to three-letter combinations of nucleotides (codons) based on Table 2.
5. Check a match for the codon from step 4 in Table 3 and compare the value with the matched color group:
   - If the color is red, go to step 6 and continue.
   - If the color is blue, use the XOR gate with all eight bits of the second pixel's blue color.
   - If the color is green, use the XOR gate with all eight bits of second pixel's green and blue colors, respectively.
6. Concatenate the resulted bits to the BM in proper order.
7. If stop condition which is eight bits of zeros is not satisfied, go to step 2. Otherwise, go to the next step.
8. Convert the binary sequence BM to message M.
9. The recovery process is completed, and the result is the original message M.

The example of the data recovery scheme works as follows: once stego image SIMG is received, two pixels are taken at a time and the LSBs of two pixels are taken which can make six binary digits. This binary number is then mapped to corresponding nucleotides and codon base on

Table 2 and then Table 3 respectively, in the example that is given in the data hiding phase was 011011 that equivalent to "GCT." In addition, the codon is checked in Table 3 and in our given example the "GCT" happens to have three other codons in the same group (GCA, GCC, GCG, GCT) for that two last bits are taken after XOR gate are applied of the last two corresponding colors of pixel 2. The process is repeated for the rest of the image pixels until the whole message sequence is retrieved.

5. RESULTS

The proposed scheme is tested and appraised using different standard measurements that are commonly used for steganography evaluation. For this purpose, the used measurements are hiding capacity and visual quality. As for the used tools in the proposed method, a project is developed in C# language, and a reasonable number of datasets have been tested. However, mathematical models were used, such as the likelihood probability, where each value is equally likely to appear in image LSB pixels, and thus bias is avoided when results are calculated.

5.1. Hiding Capacity

Hiding capacity refers to the number of bits from a secret message that can be embedded per pixel, and it is measured by bits per pixels (bpp) (Shiu et al., 2010). The higher the bpp, the more information can be embedded in the cover image. However, high bpp means more distortion on the cover image; therefore, maintaining the visual quality and other security measurements should be considered when the bpp increases. Even though the likelihood of bits distribution may vary from one image to another, for measurement purposes, it is assumed that all values are equally likely to appear in the LSB of RGB colors. In the proposed method, two pixels are taken, and the LSBs are taken for all three colors in sequence, which results in six bits sequence, and they are mapped to different codons on the basis of Table 2, which is then converted to amino acids. Since there are 64 possible values, the probability of any value that can appear is 1/64. The distribution probability of the codons from Table 3 is 4/64 for red color codons, where no bits can be embedded. For blue color, the probability is 28/64, and one bit can be embedded in the cover image. Lastly, for the green color, the probability is 32/64, where two bits can be embedded in the cover image. By calculating the probabilities, the hiding capacity is 1.438 for two pixels, based on the below equation and the bpp is 0.719.

$$Hiding\ capacity = \frac{0\times\frac{4}{64}+1\times\frac{28}{64}+2\times\frac{32}{64}}{2} = \frac{1.438}{2} = 0.719 \quad (1)$$

5.2. Visual Quality

Visual quality refers to the changes that occur in the cover image that change its quality and how much these changes are unnoticeable by human eyes; in other words, it refers to distortion or the visual modification of the cover image. The two metrics that are widely used are mean squared error (MSE) and peak signal-to-noise ratio (PSNR) (Hussain et al., 2018).

The MSE measures the error or changes between the original cover image and the cover image after embedding stego message, and it is defined as

$$MSE = \frac{1}{H\times W}\sum_{y=1}^{W}\sum_{x=1}^{H}(IMG(x,y) - SIMG(x,y))^2 \quad (2)$$

where H and W are height and width, respectively, of the cover image IMG and stego image SIMG, while IMG (x,y) and SIMG (X,Y) denotes the values the pixel's RGB of both cover and stego images with specified coordinates of height and width. The lower the MSE value, the better it is, because it means less distortion. For calculating the MSE, the assumption is that each value is equally likely to appear in the LSBs, and thus, the hiding capacity will be considered. Because there is a 50% chance that the value LSB will alter when the LSB is not equal to the stego bit, the probability of each pixel to alter roughly equal to 0.36. Therefore, the MSE for the proposed method is 0.36.

The PSNR measures the ratio between the maximum intensity value of pixel for the cover image and the MSE difference of both cover and stego images. The PSNR is an indicator to find the real visual differences between the cover and the stego image because it depends on color intensity. MSE for two images may have the same value while one image has a lower power of magnitude and the other one has a larger power of magnitude; thus, the lower magnitude power image will have more distortion even though both images have the same MSE. For that, PSNR is used, and a larger value of PSNR indicates less distortion and the quality of image maintained, which can be undetectable by human eyes. And it calculated as follow:

$$PSNR = 10 \times \log_{10}\left(\frac{Max^2}{MSE}\right) = 10 \times \log_{10}\left(\frac{255^2}{\left(\frac{0.36}{3}\right)}\right) = 57.33 \quad (3)$$

The MSE is equal to 0.36, as calculated from the MSE equation. Division by three represents the RGB colors because RGB is used instead of grayscale colors. Max is the maximum pixel color intensity, which is equal to 255. The result of the proposed method is equal to 57.33 dB, and it performs better than a number of famous methods referenced in Table 4.

The PSNR results of the proposed method compared with similar work are indicated in Table 4, and the proposed scheme has higher a PSNR with the same embedding capacity of around 0.7 bit per pixel.

Table 4: PSNR value compared with similar methods

Method	**PSNR for embedding capacity at (0.7)**
LSB++ (Ghazanfari, Ghaemmaghami, and Khosravi, 2011)	55.82
RHTF-based LSB (Lou and Hu, 2012)	55.36
RHTF-based LSB++ (Nag et al., 2016)	55.44
Pattern bits shuffling (Muhammad, Ahmad, Farman, and Jan, 2016)	~52
Proposed method	57.33

5.3. Cracking probability

Even though image steganography is mostly about the visualization of the image and to what extent the image has been distorted, when attackers have a suspicion about the image, then the image should have a high cracking probability to make it hard for attackers to retrieve hidden data. In short, cracking probability is the probability of attackers making a correct guess to discover a hidden message. The proposed method is simple to accomplish, yet, the cracking probability of the presented method is high. To precisely retrieve the hidden data, attackers need the following information:

First: the number of pixels used at a time, because one pixel is skipped at a time and the probability to detect that is 1/(n), where n is the number of pixels in the image

Second: the number of bits and their position used in the XOR process is 1/(8!)

Third: the number of LSBs used in embedding process depends on the value of the LSB and can vary, having zero to two bits. In case there are no bits to hide, the probability is to check the whole image and find the four codon values that are not embedding bits, and that is $1/(n*(64)^4)$. When hiding one bit, then the probability of finding 28 codons that are used for hiding one bit and guessing its position in the pixel is $1/(n*3*(64)^{28})$. When hiding two bits, then the probability of finding 32 codons that are used for hiding two bits and guessing their place in the pixel is $1/(n*3*(64)^{32})$.

Fourth: to successfully guess the original message among all possible values, it requires the exponent value of the binary message length, which is $1/(2^m)$.

Thus the cracking probability is equal to:

$$Cracking\ probability = \frac{1}{n} \times \frac{1}{8!} \times \frac{1}{(n \times (64)^4) \times (n \times 3 \times (64)^{28}) \times (n \times 3 \times (64)^{32})} \times \frac{1}{2^m} \quad (4)$$

6. CONCLUSION

In this paper, a new data hiding scheme has been proposed. Unlike traditional techniques, the proposed method is based on the biological functionality of DNA sequence and not on a mathematical model alone. The proposed method uses the translation process characteristics of DNA sequence, in which codons in the sequence are translated into amino acids. The proposed method provides an improvement over traditional LSB, and it cannot be detected easily because it does not inject the stego bits directly to all LSBs of the cover image. In addition, the distortion on the cover image is very low and is not detectable easily. The results from the PSNR and MSE are indicators of the effectiveness of the proposed method.

REFERENCES

Abdo, A. M., Sabry, A., & Abdullah, A.A. (2018). A new message encryption method based on amino acid sequences and genetic codes. *International Journal of Advanced Computer Science and Applications*. 9(8).

Amoroso, E. G. (1999). *Intrusion detection: An introduction to Internet Surveillance, Correlation, Traps, Trace Back, and Response* (1st ed). Sparta, N.J: Intrusion.Net Books.

Cheddad, A., Condell, J., Curran, K., & Mc Kevitt, P. (2010). Digital image steganography: survey and analysis of current methods. *Signal Processing*. 90(3), 727-752. https://doi.org/10.1016/j.sigpro.2009.08.010

Comer, D. (2019). *The Internet book: Everything You Need to Know About Computer Networking and How the Internet Works* (Fifth edition). Boca Raton: CRC Press, Taylor & Francis Group.

De Silva, P. Y., & Ganegoda, G. U. (2016). New trends of digital data storage in DNA. *BioMed Research International*. 1-14.

Denemark, T., Boroumand, M., & Fridrich, J. (2016). Steganalysis features for content-adaptive JPEG steganography. *IEEE Transactions on Information Forensics and Security*. 11(8), 1736-1746.

Djebbar, F., Ayad, B., Meraim, K. A., & Hamam, H. (2012). Comparative study of digital audio steganography techniques. *EURASIP Journal on Audio, Speech, and Music Processing*. 2012(1), 25.

Ghazanfari, K., Ghaemmaghami, S., & Khosravi, S. R. (2011). LSB^{++}: An improvement to LSB^{+} steganography. *TENCON 2011 - 2011 IEEE Region 10 Conference*. 364-368.

Gupta, Ankur Goyal, S., & Bhushan, B. (2012). Information hiding using least significant bit steganography and cryptography. *International Journal of Modern Education and Computer Science*, 4(6), 27-34.

Hafeez, I., Khan, A., & Qadir, A. (2014). DNA-LCEB: A high-capacity and mutation-resistant DNA data-hiding approach by employing encryption, error correcting codes, and hybrid twofold and fourfold codon-based strategy for synonymous substitution in amino acids. *Medical & Biological Engineering & Computing*. 52(11), 945-961.

Huang, Y.H., Chang, C.C., & Wu, C.Y. (2014). A DNA-based data hiding technique with low modification rates. *Multimedia Tools and Applications*. 70(3), 1439-1451.

Hussain, M., Wahab, A. W. A., Idris, Y. I. B., Ho, A. T. S., & Jung, K.H. (2018). Image steganography in spatial domain: a survey. *Signal Processing: Image Communication*. 65, 46-66.

Jorde, L. B., Carey, J. C., & Bamshad, M. J. (2016). *Medical genetics* (Fifth edition). Philadelphia, PA: Elsevier.

Leier, A., Richter, C., Banzhaf, W., & Rauhe, H. (2000). Cryptography with DNA binary strands. *Biosystems*. *57*(1), 13-22.

Lou, D.C., & Hu, C.H. (2012). LSB steganographic method based on reversible histogram transformation function for resisting statistical steganalysis. *Information Sciences*. 188, 346-358.

Mstafa, R. J., & Elleithy, K. M. (2017). Compressed and raw video steganography techniques: a comprehensive survey and analysis. *Multimedia Tools and Applications*, 76(20), 21749-21786.

Mstafa, R. J., Elleithy, K. M., & Abdelfattah, E. (2017). Video steganography techniques: Taxonomy, challenges, and

future directions. *2017 IEEE Long Island Systems, Applications and Technology Conference (LISAT)*, 1–6.

Muhammad, K., Ahmad, J., Farman, H., & Jan, Z. (2016). A new image steganographic technique using pattern based bits shuffling and magic LSB for grayscale images. *ArXiv:1601.01386* [Cs]. Retrieved from http://arxiv.org/abs/1601.01386

Nag, A., Choudhary, S., Basu, S., & Dawn, S. (2016). An image steganography scheme based on LSB++ and RHTF for resisting statistical steganalysis. *IEIE Transactions on Smart Processing and Computing*. 5(4), 250-255.

Nussbaum, R. L., McInnes, R. R., & Willard, H. F. (2016). *Thompson & Thompson Genetics in Medicine* (Eighth edition). Philadelphia: Elsevier.

Shiu, H. J., Ng, K. L., Fang, J. F., Lee, R. C. T., & Huang, C. H. (2010). Data hiding methods based upon DNA sequences. *Information Sciences*, 180(11), 2196-2208. https://doi.org/10.1016/j.ins.2010.01.030

ur Rehman, A., Liao, X., Kulsoom, A., & Abbas, S. A. (2015). Selective encryption for gray images based on chaos and DNA complementary rules. *Multimedia Tools and Applications*. 74(13), 4655-4677.

Wayner, P. (2009). *Disappearing Cryptography: Information Hiding: Steganography & Watermarking* (3rd ed). Amsterdam ; Boston: Morgan Kaufmann Publishers.

Wu, H., Dugelay, J.L., & Cheung, Y. (2008). A data mapping method for steganography and its application to images. In K. Solanki, K. Sullivan, & U. Madhow (Eds.), *Information hiding.* 5284, 236-250.

Yurke, B., Turberfield, A. J., Mills, A. P., Simmel, F. C., & Neumann, J. L. (2000). A DNA-fuelled molecular machine made of DNA. *Nature*. 406(6796), 605-608